7.6 A 4Gb 2b/cell NAND Flash Memory with Embedded 5b BCH ECC for 36MB/s System Read Throughput

R. Micheloni[1], R. Ravasio[1], A. Marelli[1], E. Alice[2], V. Altieri[2], A. Bovino[2], L. Crippa[1], E. Di Martino[2], L. D'Onofrio[2], A. Gambardella[2], E. Grillea[2], G. Guerra[2], D. Kim[3], C. Missiroli[1], I. Motta[1], A. Prisco[2], G. Ragone[1], M. Romano[2], M. Sangalli[1], P. Sauro[2], M. Scotti[1], S. Won[3]

[1]STMicroelectronics, Agrate Brianza, Italy
[2]STMicroelectronics, Arzano, Italy
[3]Hynix Semiconductor, Icheon, Korea

The development of multilevel cell (MLC) NAND Flash architecture allows a dramatic increase in memory density [1]. As cost per bit decreases, the ever-growing portable mass-storage market (USB keys, MP3 players, and so on) may be invaded by MLC devices. MLC data retention and cycling are inferior to those of single bit cell (SBC) NAND Flash, due to closer margins between distributions. For reliability reasons, SBC Flash requires an ECC [2] able to correct one error per page while more bits must be corrected in the case of MLC.

An SBC memory that integrates one error Hamming ECC has been previously reported [3]. In this paper, a 4Gb MLC NAND Flash that integrates an implementation of a 5-errors BCH code over 2102B data field using 10 parity bytes is reported. Any software implementation of BCH on commercial microprocessors results in a read-throughput degradation to some MB/s; the main reason being the lack of division and multiplications operators defined in a Galois field ($GF(2^{15})$). The ALU of the embedded microcontroller in the proposed design includes these operations and through a dedicated logic circuitry it allows the external controller to download data at 36MB/s (cache read, x16). This eliminates the need for any special hardware to the external controller, which directly impacts the cost of the application.

The operating frequency of the ECC core is >25MHz. If no error occurs, ECC time overhead for program and read operations is 500ns. A 4 bit-basis corrector is used in case of 2 to 5 errors with 250µs time overhead while a 16 bit-basis one manages the more likely single error in 34µs, ensuring a good tradeoff between area and latency. ECC area overhead is 1.3mm^2, <1% of the chip area; no extra BLs are added. The average current draw is <1mA.

Many fields with flexible length - 1B to 2102B - may be allocated into a memory page. This feature improves read throughput when the application downloads a file split on different pages.

Figure 7.6.1 shows a micrograph of the chip designed in a 90nm technology. The memory array is composed of 2k blocks. Each block, split into left and right side, is made up of 128 pages of (2k + 64)B each.

Figure 7.6.2 shows the functional blocks involved in BCH encoding and decoding as well as the chosen polynomials. The generator polynomial (g) is used to compute the data-field parity. The minimal polynomials (ϕ_1, ϕ_3, ϕ_5, ϕ_7, ϕ_9) are used to compute 5 syndromes (15b each, to correct up to 5 errors) associated to read data. The error condition is detected when the syndromes calculated from the data field and its parity bits are different from zero. In this case, Berlekamp - Massey algorithm is used to compute the error locator polynomial. The error positions are found by Chien machine, exploiting polynomial roots search in $GF(2^{15})$. As the error positions are known, the external controller can easily correct the read data.

Figure 7.6.3 shows how the BCH structure is implemented in the 4Gb MLC. On data input, 75 parity bits are computed on word-basis hardware and stored in 10B that will be appended to the end of the data field for page buffers loading. On data output, syndromes are computed using an architecture similar to the parity one. Status-register reading allows detection of any error; after that, get-error-position (GEP) function may be invoked. At this point, the internal microcontroller used to perform read and write algorithms acquires the syndromes stored into the registers and performs Berlekamp decision-tree algorithm that is optimized to manage more likely errors in shorter time. The algorithm is composed of <500 instructions and it needs eight 16b registers. The last decoding block is the Chien FSM that searches for the error locator polynomial roots, which are the error positions. Since the calculation of the error position is the most time-consuming operation, parallelism should be exploited. However, the complexity and area overhead of Chien FSM grow dramatically as the parallelism increases. Single error is more likely to happen than multiple errors. Since in this case Berlekamp algorithm finds only the first coefficient of the first-degree error locator polynomial, in the proposed solution, 2 different Chien cores are implemented: a simplified Chien core finds single-error position while another one manages 2 to 5 errors cases. Both cores share common structures even if in Fig. 7.6.3 they are shown as separated cores. Figure 7.6.4 highlights the worst-case latency of the proposed approach: 60µs for single error and 250µs for 5 errors.

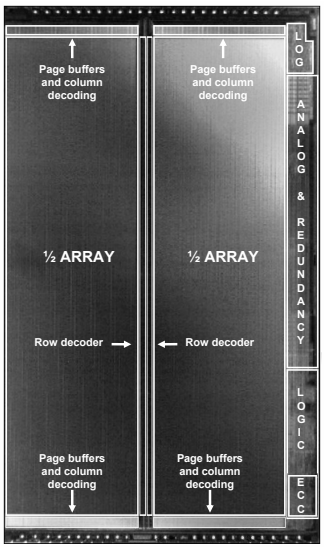
The hardware required by the sequential BCH structure (parity, syndromes and Chien machines) is based on a linear shift register (LSR). Starting from this implementation, Fig. 7.6.5 shows the details of the parity machine design. In the standard approach, parity bits are computed as the remainder of a serial polynomial division between data field and generator polynomial. The computation is easily performed by an LSR composed of 75 D-FF and adders represented by XOR functions. XORs take place where there is a non-zero coefficient in the generator polynomial of the code. This sequential implementation does not fit 8b natural parallelism of the device. The circuit could be unrolled to fit, thus reducing latency time for calculations. However, the length of the path makes this implementation slow and power consuming. A different architecture is proposed starting from the unrolled one, describing register inputs as linear functions of register output and 8b inputs. This description can be efficiently synthesized, sharing repeated expressions to minimize area overhead. The result is a short-path implementation with reduced latency time and power consumption.

The flexibility introduced in terms of number and size of data fields allows an efficient file management. In spite of using a serial ECC like BCH, it is possible to read file pointers without waiting for reading the whole page (2102B). This is particularly useful when more pages are read. As shown in Fig. 7.6.6, this flexibility is managed by the command interface.

A summary table of the performances of the BCH implemented in the 4Gbit MLC chip is sketched in Fig. 7.6.7. The GEP time is proportional to the size of the data field: in case of 2 errors, 90% reduction in latency time is achieved as data field decreases from 2048B to 16B.

Acknowledgments:
We thank C. Golla, M. Carrera and the layout team, A. Aldarese, G. Salsano and the ST product team, ST CAD team, Hynix and ST R&D for their support.

References:
[1] D-S. Byeon,, et al., "An 8Gb Multi-Level NAND Flash Memory with 63nm STI CMOS Process Technology," *ISSCC Dig. Tech. Papers*, pp. 46-47, Feb., 2005.
[2] S. B. Wicker, Error Control Systems for Digital Communication and Storage, Prentice Hall, 1995.
[3] S. Tanzawa, et al., "A Compact on-chip ECC for Low Cost Flash Memories," *IEEE J. Solid-State Circuits*, vol. 32, pp 662-669, May, 1997.



Technology	**90nm CMOS**
	3 metal (1W, 2Al)
Cell size	**0.044µm²**
Chip area	**140mm²**
Organization	**2112 x 128 pages x 2kblocks x 8**
	1056 x 128 pages x 2kblocks x 16
Cycle time	**50ns**
Read time	**35µs (average)**
Cache Read	**available**
I/O	**x8 - x16**

Figure 7.6.1: Die micrograph.

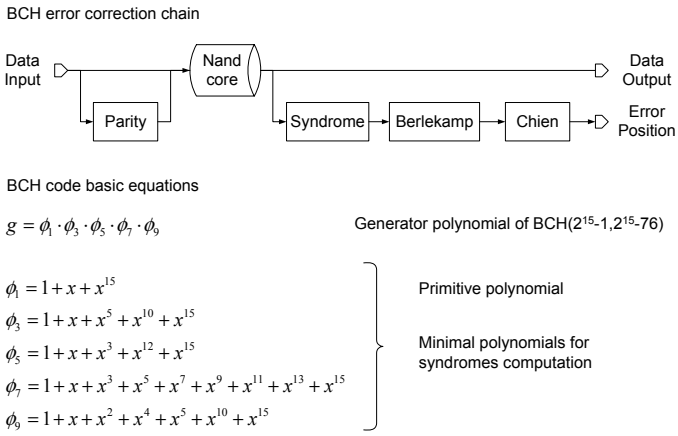


Figure 7.6.2: BCH main functional blocks.

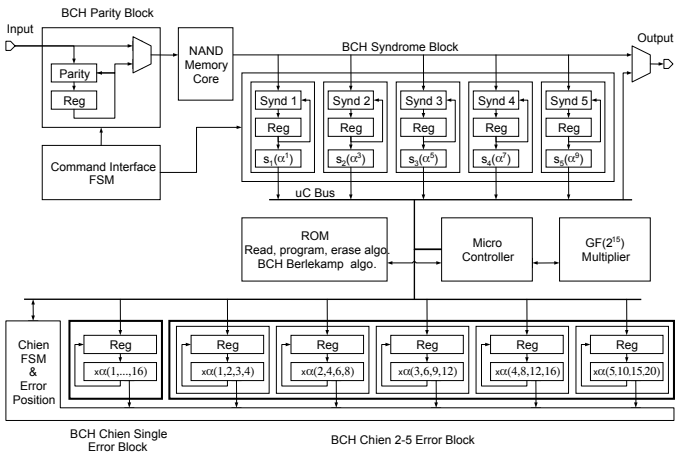


Figure 7.6.3: BCH structure.

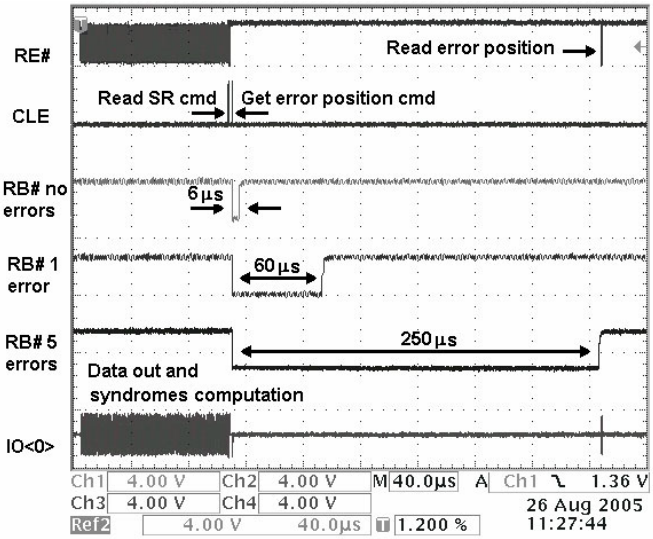


Figure 7.6.4: Worst-case error-position computation overhead.

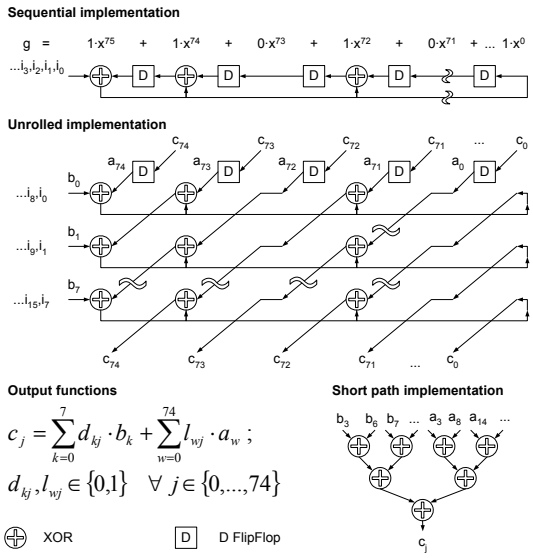


Figure 7.6.5: BCH parity details.

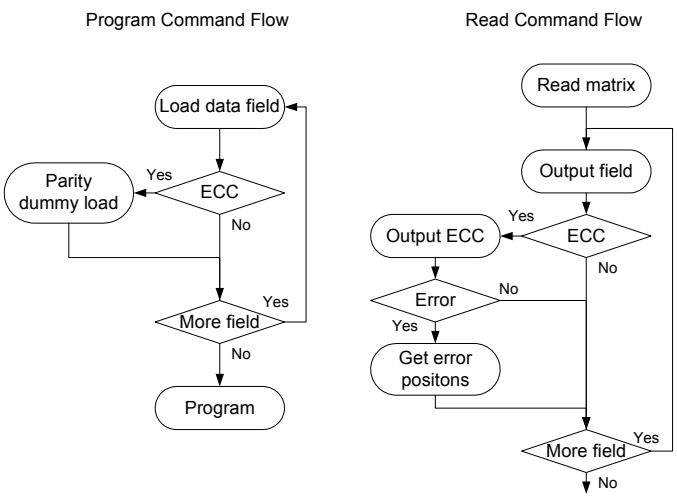


Figure 7.6.6: Memory command interface.

Continued on Page 644

Error correction code	BCH(32767,32692)			
Data field size	1B to 2102B			
# of field in a page	User defines number of fields in a page and their length			
Parity bits overhead	10B/Field			
Parity time overhead	500ns (10 write cycle for dummy parity load)			
Syndrome time overhead	500ns (10 read cycle for parity read)			
Get error position time overhead (μs) (Berlekamp and Chien)		Field dimension		
		16B	512B	2048B
	1 error (typ)	8.7	14.9	34.1
	1 error (max)	8.9	21.3	59.7
	2 errors (max)	20.1	69.7	223
	3 errors (max)	29.2	78.8	232
	4 errors (max)	38.3	87.9	241
	5 errors (max)	46.7	96.3	250
ECC area overhead	1.3mm^2			
ECC current draw	<1mA			
ECC max operating freq.	25MHz			

Figure 7.6.7: ECC performances.

